

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S96	15	((function method) adj call) same table)and S92	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 10:56
S95	43	((function method) adj call) and S92	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 10:56
S94	27	(function adj call) and S92	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 10:55
S93	3	vtbl and S92	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 10:55
S92	110	717/165.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 10:54
S91	52	"133" and S89	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 10:50
S90	0	"magnetic disk interface" and S89	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 09:33
S89	1040	"2000".py. and S88	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 09:32
S88	12820	microsoft.as.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/02/01 09:31
S87	2	("5793374").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/02/01 09:27

S86	2	("6077313").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/02/01 09:17
-----	---	-----------------	---------------------------------------------------------	----	-----	------------------



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☐ The ACM Digital Library ☒ The Guide

memoization

SEARCH

THE GUIDE TO COMPUTING LITERATURE


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Term used **memoization**Found **335** of **846,556**

Sort results by

relevance

Display results

expanded form

[Save results to a Binder](#)
[Search Tips](#)
☐ Open results in a new window
Try an [Advanced Search](#)Try this search in [The Digital Library](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐1 [Selective memoization](#)

Umut A. A. Acar, Guy E. Blelloch, Robert Harper

January 2003 **ACM SIGPLAN Notices , Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 38 Issue 1Full text available: [pdf\(152.81 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We present a framework for applying memoization selectively. The framework provides programmer control over equality, space usage, and identification of precise dependences so that memoization can be applied according to the needs of an application. Two key properties of the framework are that it is efficient and yields programs whose performance can be analyzed using standard techniques. We describe the framework in the context of a functional language and an implementation as an SML library. Th ...

Keywords: memoization, performance, programmer controlled, selective2 [Developing a tool for memoizing functions in C++](#)

Paul McNamee, Marty Hall

August 1998 **ACM SIGPLAN Notices**, Volume 33 Issue 8Full text available: [pdf\(619.00 KB\)](#) Additional Information: [full citation](#), [citations](#), [index terms](#)3 [Technical correspondence: Techniques for automatic memoization with applications to context-free parsing](#)

Peter Norvig

March 1991 **Computational Linguistics**, Volume 17 Issue 1Full text available: [pdf\(410.96 KB\)](#)[Publisher Site](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

It is shown that a process similar to Earley's algorithm can be generated by a simple top-down backtracking parser, when augmented by automatic memoization. The memoized parser has the same complexity as Earley's algorithm, but parses constituents in a different order. Techniques for deriving memo functions are described, with a complete implementation in Common Lisp, and an outline of a macro-based approach for other languages.

4 [Squibs and discussions: Memoization in top-down parsing](#)

Mark Johnson

September 1995 **Computational Linguistics**, Volume 21 Issue 3



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
Search: ☐ The ACM Digital Library ☒ The Guide

memoization virtual

SEARCH

THE GUIDE TO COMPUTING LITERATURE

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **memoization virtual**

Found 30,106 of 846,556

Sort results
by

relevance

Display
results

expanded form

[Save results to a Binder](#) [Search Tips](#)☐ Open results in a new windowTry an [Advanced Search](#)Try this search in [The Digital Library](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Developing a tool for memoizing functions in C++](#)

Paul McNamee, Marty Hall

August 1998 **ACM SIGPLAN Notices**, Volume 33 Issue 8Full text available: [pdf\(619.00 KB\)](#) Additional Information: [full citation](#), [citations](#), [index terms](#)**2** [A performance analysis of hierarchical shortest path algorithms](#)

A. Fetterer

November 1997 **Proceedings of the 9th International Conference on Tools with Artificial Intelligence (ICTAI '97)**Full text available: [Publisher Site](#) Additional Information: [full citation](#), [abstract](#)

Abstract: The shortest-path problem is an essential component of many applications including Advanced Traveler Information Systems (ATIS), computer networks, etc. A hierarchical routing algorithm decomposes the original graph into a set of fragment graphs and also into a boundary graph which summarizes the fragment graphs. A fully memoized hierarchical routing algorithm pre-computes and stores the shortest-path data structure and the shortest-path-cost data structure for the graph fragments, as ...

Keywords: Advanced Traveler Information Systems, Twin Cities metropolitan road-map, boundary graph, computation time, computer networks, fragment graphs, fully memoized hierarchical routing algorithm, hierarchical shortest path algorithms, metropolitan area networks, performance analysis, real graph, shortest-path data structure, storage overhead

3 [An efficient profile-analysis framework for data-layout optimizations](#)

Shai Rubin, Rastislav Bodík, Trishul Chilimbi

January 2002 **ACM SIGPLAN Notices**, **Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 37 Issue 1Full text available: [pdf\(245.74 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Data-layout optimizations rearrange fields within objects, objects within objects, and objects within the heap, with the goal of increasing spatial locality. While the importance of data-layout optimizations has been growing, their deployment has been limited, partly because they lack a unifying framework. We propose a parameterizable framework for data-layout optimization of general-purpose applications. Acknowledging that finding an optimal layout is not only NP-hard, but also poorly approxima ...

4 [A member lookup algorithm for C++](#)

G. Ramalingam, Harini Srinivasan



ruf memoization

Search

[Advanced Scholar Search](#)
[Scholar Help](#)

Scholar

Results 1 - 10 of about 41 for ruf memoization. (0.06 seconds)

[PS] Readings in Context Sensitive Interprocedural Dataflow Analysis

B Concepts, V Predication, PT Functions - [View as HTML](#)

... Erik Ruf. April 2, 2001. 1 Basic Concepts. ... [EGH94] full inlining of call graph (modulo recursion, **memoization**). [Shi88] stack-depth based contours (n-CFA). ...

cs.washington.edu

[PS] How to have your cake and eat it, too: Self-applicable online partial evaluation

M Sperber - [View as HTML](#) - Cited by 2

Page 1. How To Have Your Cake and Eat It, Too: Self-Applicable Online Partial Evaluation. Michael Sperber. Wilhelm-Schickard-Institut ...

Danvy et al.[32].(To appear) - www-pu.informatik.uni-tuebingen.de

[PDF] RG: A Case-Study for Aspect-Oriented Programming

A Mendhekar, G Kiczales, J Lamping - [View as HTML](#) - Cited by 42

... The output program will have all the properties with respect to **memoization**, fusion and memory ... 8. Lamping, J., Kiczales, G., Rodriguez Jr., LH, and Ruf, E. An ...

Submitted to OOPSLA 97 for possible publication, 1997 - isis.vanderbilt.edu - parc.com - parc.xerox.com - ftp.funet.fi - [all 10 versions](#) »

[PDF] Specializing shaders

B Guenter, TB Knoblock, E Ruf - Cited by 44

Page 1. 1. Specializing Shaders. Brian Guenter, Todd B. Knoblock, Erik Ruf *.

Microsoft Research. Abstract. We have developed a system ...

Computer Graphics Proceedings, Annual Conference Series - portal.acm.org - graphics.lcs.mit.edu - research.microsoft.com - research.microsoft.com - [all 6 versions](#) »

[PS] The generation of a higher-order online partial evaluator

P Thiemann, R Gluck - [View as HTML](#) - Cited by 11

... generation provides for a good separation of concerns: the underlying oine partial evaluator (Similix 4) handles code generation and **memoization**, while the ...

Fuji International Workshop on Functional and Logic ..., 1995 - informatik.uni-freiburg.de

[PDF] Program analysis with partial transfer functions

B Murphy, M Lam - Cited by 2

Page 1. Program Analysis with Partial Transfer Functions* Brian R. Murphy Monica

S. Lam Computer Systems Laboratory Stanford University ...

Proceedings of the Workshop on Partial Evaluation and ... - portal.acm.org - suif.stanford.edu - portal.acm.org

[PDF] Characterizing coarse-grained reuse of computation

SS Sastry, R Bodik, JE Smith - [View as HTML](#) - Cited by 4

... computation reuse in programs at coarse granularities, and in determining the relative appli- cability of specialization and **memoization**, two commonly used ...

3rd ACM Workshop on Feedback-Directed and Dynamic ..., 2000 - ece.wisc.edu - ece.wisc.edu - cs.wisc.edu - eecs.harvard.edu - [all 5 versions](#) »

[PDF] Preserving information during online partial evaluation

E Ruf, D Weise - [View as HTML](#) - Cited by 3

... Erik Ruf and Daniel Weise. Technical Report: CSL-TR-92-517 also FUSE Memo 92-8

Erik Ruf is funded byanAT&TFoundation Ph.D. Scholarship. i . Page 2. ...

Technical Report CSL-TR-92-517, Computer Systems Laboratory, ... - reports.stanford.edu



ruf memoization indirect

Search

[Advanced Scholar Search](#)
[Scholar Help](#)

Scholar

Results 1 - 10 of 10 for ruf memoization indirect. (0.15 seconds)

[PS] Connection analysis: A practical interprocedural heap analysis for C

R Ghiya, LJ Hendren - [View as HTML](#) - Cited by 56

... in the. 8. Page 9. program and are directly or indirectly (through an **indirect** reference) accessible from. the function. Names are naturally ...

International Journal of Parallel Programming, 1996 - [cs.colorado.edu](#) - [sable.mcgill.ca](#) - [www-acaps.cs.mcgill.ca](#) - [portal.acm.org](#) - [all 7 versions](#) »

[PDF] Estimating the impact of scalable pointer analysis on optimization

M Das, B Liblit, M Fahndrich, J Rehof - Cited by 22

... The number of implicit assignments is linear even in the presence of **indirect** calls, because there is a single signature for all possible target ... **Memoization**. ...

8th International Symposium on Static Analysis, 2001 - [springerlink.com](#) - [cs.berkeley.edu](#) - [portal.acm.org](#) - [research.microsoft.com](#) - [all 7 versions](#) »

[PS] Hand-writing program generator generators

L Birkedal, M Welinder - [View as HTML](#) - Cited by 24

... presenting a program generator generator for a skeletal language, and we argue that it is not more difficult to use the direct approach than the **indirect** approach ...

Proceedings of the 6th International Symposium on ..., 1994 - [www-2.cs.cmu.edu](#) - [itu.dk](#) - [it-c.dk](#) - [portal.acm.org](#) - [all 7 versions](#) »

[PS] Is it a tree, a DAG, or a cyclic graph

R Ghiya, L Hendren - [View as HTML](#) - Cited by 26

Page 1. To appear in Proceedings of the Twenty Third ACM SIGPLAN{SIGACT Symposium on Principles of Programming Languages, St. Petersburg ...

Proc. 23rd ACM POPL, 1996 - [www-acaps.cs.mcgill.ca](#) - [amber.cs.umd.edu](#) - [cs.umd.edu](#) - [cs.cornell.edu](#) - [all 7 versions](#) »

[PDF] Interprocedural pointer alias analysis

M Hind, M Burke, P Carini, JD Choi - Cited by 60

... function pointers. PCG construction is performed in an optimistic manner; all **indirect** calls are initially assumed to call no functions. ...

ACM Transactions on Programming Languages and Systems, 1999 - [portal.acm.org](#) - [newit.gsu.unibel.by](#) - [portal.acm.org](#) - [research.ibm.com](#)

[PDF] Value Profiling for Instructions and Memory Locations

UT Report, PT Feller, PB Calder, CPD Tullsen, PJ ... - [View as HTML](#)

... Knoblock and. **Ruf** 25] used a form of staging analysis and annotations to guide data specialization. ... These **indirect** function calls are virtual function calls,. ...

[cs.ucsd.edu](#) - [charlotte.ucsd.edu](#) - [cs.ucsd.edu](#)

[PS] Lightweight Languages for InteractiveGraphics

S Draves - [View as HTML](#) - Cited by 6

... All tests, loops, and **indirect** calls have been eliminated; the remaining code can be scheduled more easily and will run faster. ...

CMUCS-95-148 - [cs.cmu.edu](#) - [portal.acm.org](#)

[PS] Language and Compiler Support for Dynamic Code Generation

MA Poletto - [View as HTML](#) - Cited by 1

Page 1. Language and Compiler Support for. Dynamic Code Generation. by. Massimiliano A. Poletto. SB, Massachusetts Institute of Technology ...